ABSTRACT

A computing architecture and software techniques are described which modifies the basic sequential instruction fetching mechanism of a processor by separating a program's control flow from its functional execution flow. A compiled sequential HLL program's static control structures are analyzed and a separate program based on its own unique instructions is created that primarily generates addresses for the selection of functional execution instructions. The original program is now represented by an instruction fetch program and a set of function/logic execution instructions. This basic split allows multiple instruction addresses to be generated in parallel to access multiple instruction memories. These multiple instruction memories contain only the function/logic instructions of the program and no control structure operations such as branches or calls. All the original program's control instructions are split from the original program and used to create the instruction addressing program. This approach allows a variable number of instructions to be issued in parallel whenever the program can allow for it. The instructions of this approach are referred to herein as assembled variable length instructions or AVLIs. Alternative techniques are provided that deal with conditional and unconditional branches. In addition, all or a majority of duplicate function/logic instructions can be removed relying on a single copy or a small number of copies to be stored and referenced as needed by the control program based on architecture features so that overall instruction storage can be reduced.